Learning-augmented Online Minimization of Age of Information and Transmission Costs

Zhongdong Liu, Keyuan Zhang, Bin Li, Yin Sun, Y. Thomas Hou, and Bo Ji

***Abstract*—We consider a discrete-time system where a resource-constrained source (e.g., a small sensor) transmits its time-sensitive data to a destination over a time-varying wireless channel. Each transmission incurs a fixed transmission cost (e.g., energy cost), and no transmission results in a staleness cost represented by the *Age-of-Information*. The source must balance the tradeoff between transmission and staleness costs. To address this challenge, we develop a robust online algorithm to minimize the sum of transmission and staleness costs, ensuring a worst-case performance guarantee. While online algorithms are robust, they are usually overly conservative and may have a poor average performance in typical scenarios. In contrast, by leveraging historical data and prediction models, machine learning (ML) algorithms perform well in average cases. However, they typically lack worst-case performance guarantees. To achieve the best of both worlds, we design a learning-augmented online algorithm that exhibits two desired properties: (i) *consistency*: closely approximating the optimal offline algorithm when the ML prediction is accurate and trusted; (ii) *robustness*: ensuring worst-case performance guarantee even ML predictions are inaccurate. Finally, we perform extensive simulations to show that our online algorithm performs well empirically and that our learning-augmented algorithm achieves both consistency and robustness.**

***Index Terms*—Age-of-Information, transmission cost, online algorithm, learning-augmented algorithm.**

I. Introduction

In recent years, we have witnessed the swift and remarkable development of the Internet of Things (IoT), which connects billions of entities through wireless networks [1]. These entities range from small, resource-constrained sensors (e.g., temperature sensors and smart cameras) to powerful smartphones. Among various IoT applications, one most important categories is real-time IoT application, which requires timely information updates from the IoT sensors. For example, in industrial automation systems [2], [3], battery-powered IoT sensors are deployed to provide data for monitoring equipment health and product quality. On the one hand, IoT sensors are usually small and have limited battery capacity, and thus frequent transmissions drain the battery quickly; on the other hand, occasional transmissions render the information at the controller outdated, leading to detrimental decisions. In addition, wireless channels can be unreliable due to potential channel fading, interference, and the saturation of wireless networks if the traffic load generated by numerous sensors is high [4]. Clearly, under unreliable wireless networks, IoT sensors must transmit strategically to balance the tradeoff between transmission cost (e.g., energy cost) and data freshness. Other applications include smart grids, smart cities, and so on.

To this end, in the first part of this work, we study the tradeoff between transmission cost and data freshness under a time-varying wireless channel. Specifically, we consider a discrete-time system where a device transmits its data to an access point over an ON/OFF wireless channel (i.e., transmissions occur only when the channel is ON). Each transmission incurs a fixed *transmission cost*, while no transmission results in a *staleness cost* represented by the *Age-of-Information (AoI)* [5], which is defined as the time elapsed since the generation time of the freshest delivered packet. To minimize the sum of transmission costs and staleness costs, we develop a robust online algorithm that achieves a competitive ratio (CR) of 3. That is, different from typical studies with stationary network assumptions, the cost of our online algorithm is at most three times larger than that of the optimal offline algorithm under the worst channel state (see the definition of CR in Section III).

While online algorithms exhibit robustness against the worst-case situations, they often lean towards excessive caution and may have a subpar average performance in real-world scenarios. On the other hand, by exploiting historical data to build prediction models, machine learning (ML) algorithms can excel in average cases. Nonetheless, ML algorithms could be sensitive to disparity in training and testing data due to distribution shifts or adversarial examples, resulting in poor performance and lacking worst-case performance guarantees.

To that end, we design a novel learning-augmented online algorithm that takes advantage of both ML and online algorithms. Specifically, our learning-augmented online algorithm integrates ML prediction (a series of times indicating when to transmit) into our online algorithm, achieving two desired properties: (i) *consistency*: when the ML prediction is accurate and trusted, our learning-augmented algorithm performs closely to the optimal offline algorithm, and (ii) *robustness*: even when the ML prediction is inaccurate, our learning-augmented algorithm still offers a worst-case guarantee.

Our main contributions are as follows.

First, we study the tradeoff between transmission cost

This research was supported in part by ONR MURI grant N00014-19-1-2621, ARO grant W911NF-21-1-0244, NSF grants CNS-2106427 and CNS-2239677, Army Research Office grant W911NF-21-1-0244, Virginia Commonwealth Cyber Initiative (CCI), Virginia Tech Institute for Critical Technology and Applied Science (ICTAS), and Nokia Corporation.

Zhongdong Liu (zhongdong@vt.edu), Keyuan Zhang (keyuanz@vt.edu), and Bo Ji (boji@vt.edu) are with the Department of Computer Science, Virginia Tech, Blacksburg, VA. Bin Li (binli@psu.edu) is with the Department of Electrical Engineering, Pennsylvania State University, University Park, PA. Yin Sun (yzs0078@auburn.edu) is with the Department of Electrical and Computer Engineering, Auburn University, Auburn, AL. Y. Thomas Hou (thou@vt.edu) is with the Bradley Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA.

and data freshness in a time-varying wireless channel by formulating an optimization problem to minimize the sum of transmission and staleness costs under an ON/OFF channel.

Second, following a similar line of analysis as in [6], we reformulate our (non-linear) optimization problem into a linear Transmission Control Protocol (TCP) acknowledgment problem [7] and propose a primal-dual-based online algorithm that achieves a CR of 3. While a similar primal-dual-based online algorithm has been claimed to achieve a CR of $e/(e-1)$ [6], their analysis of CR relies on an (unrealistic) asymptotic setting (see Remark 1).

Third, by incorporating ML predictions into our online algorithm, we design a novel learning-augmented online optimization algorithm that achieves both consistency and robustness. *To the best of our knowledge, this is the first study on AoI that incorporates ML predictions into online optimization to achieve consistency and robustness.*

Finally, we perform extensive simulations using both synthetic and real trace data. Our online algorithm exhibits better performance than the theoretical analysis, and our learning-augmented algorithm can achieve consistency and robustness.

Due to space limitations, we omit all the proofs and provide them in our online technical report [8].

II. Related Work

Since AoI was introduced in [5], it has sparked numerous studies on this topic (see surveys in [9], [10]). Among these AoI studies, two categories are most relevant to our work.

The first category includes studies that consider the joint minimization of AoI and certain costs [11]–[14]. The work of [11] studies the problem of minimizing the average cost of sampling and transmission over an unreliable wireless channel subject to average AoI constraints. In [13], the authors consider a source-monitor pair with stochastic arrival of packets at the source. The source pays a transmission cost to send the packet, and its goal is to minimize the weighted sum of AoI and transmission costs. Here the packet arrival process is assumed to follow certain distributions. Although the assumptions in these studies lead to tractable performance analysis, such assumptions may not hold in practical scenarios.

The second category contains studies that focus on non-stationary settings [6], [15]–[17]. For example, in [15], the authors proposed online algorithms to minimize the AoI of users in a cellular network under adversarial wireless channels. In these AoI works that consider non-stationary settings, the most relevant work to ours is [6], where the authors study the minimization of the sum of download costs and AoI costs under an adversarial wireless channel. A primal-dual-based randomized online algorithm is shown to have a CR of $e/(e-1)$. However, this CR is attained under an (unrealistic) asymptotic setting (see Remark 1). Instead, we propose an online primal-dual-based algorithm that achieves a CR of 3 in the non-asymptotic regime. While the AoI optimization problems under non-stationary settings have been investigated, none of them considers applying ML predictions to improve the average performance of online algorithms.

In recent works [18]–[21], researchers attempt to take advantage of both online algorithms and ML predictions, i.e., to design a learning-augmented online algorithm that achieves consistency and robustness. In the seminal work [18], by incorporating ML predictions into the online Marker algorithm, the authors can achieve consistency and robustness for a caching problem. Following [18], a large body of research works in this direction have emerged. The most related learning-augmented work to ours is [20], where the authors design a primal-dual-based learning-augmented algorithm for the TCP acknowledgment problem. In [20], the uncertainty comes from the packet arrival times, and the controller can make decisions at any time. In our work, however, the uncertainty comes from the channel states, and no data can be transmitted when the channel is OFF. Lacking the freedom to transmit data at any time makes their algorithm inapplicable.

III. System Model and Problem Formulation

System Model. Consider a status-updating system where a resource-limited device sends time-sensitive data to an access point (AP) through an unreliable wireless channel. The system operates in discrete time slots, denoted by $t = 1, 2, \ldots, T$, where T is finite and is allowed to be arbitrarily large. We use $s(t) \in \{0, 1\}$ to denote the channel state at time t, where $s(t) = 1$ means the channel is ON, allowing the device to access the AP; while $s(t) = 0$ means the channel is OFF, preventing access to the AP. The sequence of channel states over the time horizon is represented by $\mathbf{s} = \{s(1), \ldots, s(T)\}$.

At the start of each slot, the device probes to know the current channel state and decides whether to transmit its freshest data to the AP. The transmission decision at slot t is denoted by $d(t) \in \{0, 1\}$, where $d(t) = 1$ if the device decides to transmit (i.e., generates a new update and transmits it to the AP), and $d(t) = 0$ if not. A scheduling algorithm π is denoted by $\pi = \{d^\pi(t)\}_{t=1}^T$, where $d^\pi(t)$ is the transmission decisions made by algorithm π at time t. For simplicity, we use $\pi = \{d(t)\}_{t=1}^T$ throughout the paper. When the device decides to transmit and the channel is ON, it incurs a fixed *transmission cost* of $c > 1$, and the data on the AP will be successfully updated at the end of slot t; otherwise, the data on the AP gets staler.[1] To quantify the freshness of data on the AP side, we utilize a metric called *Age-of-Information (AoI)* [5], which measures the time elapsed since the freshest received update was generated. We use $a(t)$ to denote the AoI at time t, which evolves as

$$a(t) = \begin{cases} 0, & \text{if } s(t) \cdot d(t) = 1; \\ a(t-1) + 1, & \text{otherwise}, \end{cases} \tag{1}$$

where the AoI drops to 0 if the device transmits at ON slots; otherwise, it increases by 1.[2] Assuming $a(0) = 0$. To reflect the penalty when the AP does not get an update at time t, we introduce a *staleness cost* equivalent to the AoI at that time.

[1] If the transmission cost c is no larger than 1, which is less than the staleness cost (at least 1), then the optimal policy is to transmit at every ON slot.

[2] Some studies let the AoI drop to 1, wherein our analysis still holds. We let the AoI drop to 0 to make the discussion concise.

Problem Formulation. The total cost of an algorithm π is

$$C(\mathbf{s}, \pi) \triangleq \sum_{t=1}^{T} (c \cdot d(t) + a(t)), \tag{2}$$

where the first item $c \cdot d(t)$ is the transmission cost at time t, and the second item $a(t)$ is the staleness cost at time t. In this paper, we focus on the class of *online* scheduling algorithms, under which the information available at time t for making decisions includes the transmission cost c, the transmission history $\{d(\tau)\}_{\tau=1}^{t}$, and the channel state pattern $\{s(\tau)\}_{\tau=1}^{t}$, while the time horizon T and the future channel state $\{s(\tau)\}_{\tau=t+1}^{T}$ is unknown. Conversely, an *offline* scheduling algorithm has the information about the connectivity pattern $\mathbf{s}$ (and the time horizon T) beforehand.

Our goal is to develop an online algorithm π that minimizes the total cost given a channel state pattern $\mathbf{s}$:

$$\min_{d(t)} \quad \sum_{t=1}^{T} (c \cdot d(t) + a(t)) \tag{3a}$$

$$\text{s.t.} \quad d(t) \in \{0, 1\} \text{ for } t = 1, 2, \ldots, T; \tag{3b}$$

$$a(t) \text{ evolves as Eq. (1) for } t = 1, 2, \ldots, T. \tag{3c}$$

In Problem (3), the only decision variables are the transmission decisions $\{d(t)\}_{t=1}^{T}$, and the objective function is a non-linear function of $\{d(t)\}_{t=1}^{T}$ due to the dependence of $a(t)$ on $d(t)$ (see detailed discussion in our technical report [8]). This non-linearity poses a challenge to its efficient solutions. In Section IV-A, following a similar line of analysis as in [6], we reformulate Problem (3) to an equivalent TCP acknowledgment (ACK) problem, which is linear and can be solved efficiently (e.g., via the primal-dual approach [22]).

To measure the performance of an online algorithm, we use the metric *competitive ratio* (CR) [22], which is defined as the worst-case ratio of the cost under the online algorithm to the cost of the optimal offline algorithm. Formally, we say that an online algorithm π is β-*competitive* if there exists a constant $\beta \geq 1$ such that for any channel state pattern $\mathbf{s}$,

$$C(\mathbf{s}, \pi) \leq \beta \cdot OPT(\mathbf{s}), \tag{4}$$

where $OPT(\mathbf{s})$ is the cost of the optimal offline algorithm for the given channel state $\mathbf{s}$. We desire to develop an online algorithm with a CR close to 1, which implies that our online algorithm performs closely to the optimal offline algorithm.

IV. Robust Online Algorithm

In this section, we first reformulate our AoI Problem (3) to an equivalent linear TCP ACK problem. Then, this TCP ACK problem is further relaxed to a linear primal-dual-based program. Finally, a 3-competitive online algorithm is developed to solve the linear primal-dual-based program.

A. Problem Reformulation

In [6], the authors study the same non-linear Problem (3) and reformulate it to an equivalent linear problem. Following a similar line of analysis as in [6], we reformulate the non-linear Problem (3) to an equivalent linear TCP ACK Problem (5) as follows. Consider a TCP ACK problem, where the source reliably generates and delivers one packet to the destination in each slot $t = 1, 2, \ldots, T$. Those delivered packets need to be acknowledged (for simplicity, we use "acked" instead of "acknowledged" throughout the paper) that they are received by the destination, which requires the destination to send ACK packets (for brevity, we call it ACK) back to the source. We use $d(t) \in \{0, 1\}$ to denote the ACK decision made by the destination at slot t. Let $z_i(t) \in \{0, 1\}$ represent whether packet i (i.e., the packet sent at slot i) has been acked by slot t ($i \leq t$), where $z_i(t) = 1$ if packet i is not acked by slot t and $z_i(t) = 0$ otherwise. Once packet i is acked at slot t, then it is acked forever after slot t, i.e., $z_i(\tau) = 0$ for all $i \leq t$ and all $\tau \geq t$. The feedback channel is unreliable and its channel state in slot t is modeled by an ON/OFF binary variable $s(t) \in \{0, 1\}$. We use $\mathbf{s} = \{s(1), \ldots, s(T)\}$ to denote the entire feedback channel states. The destination can access the feedback channel state $s(t)$ at the start of each slot t. When the feedback channel is ON and the destination decides to send an ACK, all previous packets are acked, i.e., the number of unacked packets becomes 0; otherwise, the number of unacked packets increases by 1. We can see that the dynamic of the number of unacked packets is the same as the AoI dynamic.

We assume that there is a holding cost at each slot, which is the number of unacked packets in that slot. In addition, we also assume that each ACK has an ACK cost of c. The goal of the TCP ACK problem is to develop an online scheduling algorithm $\pi = \{d(t)\}_{t=1}^{T}$ that minimizes the total cost given a feedback channel state pattern $\mathbf{s}$:

$$\min_{d(t), z_i(t)} \quad \sum_{t=1}^{T} \left(c \cdot d(t) + \sum_{i=1}^{t} z_i(t) \right) \tag{5a}$$

$$\text{s.t.} \quad z_i(t) + \sum\nolimits_{\tau=i}^{t} s(\tau) d(\tau) \geq 1 \quad \text{for } i \leq t \text{ and } t = 1, 2, \ldots, T; \tag{5b}$$

$$d(t), z_i(t) \in \{0, 1\} \text{ for } i \leq t \text{ and } t = 1, 2, \ldots, T, \tag{5c}$$

where the first item $c \cdot d(t)$ in Eq. (5a) is the ACK cost at slot t, the second item $\sum_{i=1}^{t} z_i(t)$ in Eq. (5a) is the holding cost at slot t. Constraint (5b) states that for packet i at slot t, either this packet is not acked (i.e., $z_i(t) = 1$) or an ACK was made since its arrival (i.e., $s(\tau)d(\tau) = 1$ for some $i \leq \tau \leq t$). While Problem (5) is an integer linear problem, we demonstrate its equivalence to Problem (3) in the following.

Lemma 1. *Problem* (5) *is equivalent to Problem* (3).

Proof sketch. We can show that: (i) any feasible solution to Problem (3) can be converted to a feasible solution to Problem (5), and the total costs of these two solutions are the same; (ii) any feasible solution to Problem (5) can be converted to a feasible solution to Problem (3), and the total cost of the converted solution to Problem (3) is no greater than the total cost of the solution to Problem (5). This implies that any optimal solution to Problem (3) is also an optimal solution to Problem (5), and vice versa. Therefore, these two problems are equivalent [23, Sec. 4.1.3].

To obtain a linear program of the integer Problem (5), we relax the integer requirement to real numbers:

$$\min_{d(t),z_i(t)} \sum_{t=1}^{T}\left(c\cdot d(t)+\sum_{i=1}^{t} z_i(t)\right) \tag{6a}$$

$$\text{s.t.} \quad z_i(t)+\sum\nolimits_{\tau=i}^{t} s(\tau)d(\tau)\geq 1 \quad \text{for } i\leq t \text{ and } t=1,2,\ldots,T; \tag{6b}$$

$$d(t), z_i(t)\geq 0 \text{ for } i\leq t \text{ and } t=1,2,\ldots,T, \tag{6c}$$

which is referred to as the primal problem. The corresponding dual problem of Problem (6) is as follows:

$$\max_{y_i(t)} \quad \sum_{t=1}^{T}\sum_{i=1}^{t} y_i(t) \tag{7a}$$

$$\text{s.t.} \quad s(t)\sum_{i=1}^{t}\sum_{\tau=t}^{T} y_i(\tau)\leq c \text{ for } t=1,2,\ldots,T; \tag{7b}$$

$$y_i(t)\in[0,1] \text{ for } i\leq t \text{ and } t=1,2,\ldots,T, \tag{7c}$$

which has a dual variables $y_i(t)$ for packet i and time $t\geq i$.

B. Primal-dual Online Algorithm Description and Analysis

To solve the primal-dual Problems (6) and (7), we develop the Primal-dual-based Online Algorithm (PDOA) and present it in Algorithm 1. The input is the channel state pattern $\mathbf{s}$ (revealed in an online manner), and the outputs are the primal variables $d(t)$ and $z_i(t)$, and the dual variable $y_i(t)$. Two auxiliary variables L and M are also introduced: L denotes the time when the latest ACK was made, and M denotes the ACK marker (PDOA should make an ACK when $M\geq 1$).

PDOA is a threshold-based algorithm. Assuming that the latest ACK was made at slot L, when the accumulated holding costs since slot $L+1$ is no smaller than the ACK cost c (i.e., $M\geq 1$), PDOA will make an ACK at the next ON slot L'. Here, we call the interval $[L+1, L']$ an ACK interval. Note that PDOA updates the primal variables and dual variables only for the packets that are not acked in the current ACK interval $[L+1, L']$. More specifically, consider packet i that has not been acked by the current slot $t\in[L+1, L']$: (i) for the primal variable $z_i(t)$, if the threshold is not achieved ($M<1$) or the channel is OFF at slot t, PDOA will update $z_i(t)$ to be 1 (in Line 4 or Line 15, respectively) since packet i is not acked by slot t; (ii) for the dual variable $y_i(t)$, if packet i is not the last packet in the current ACK interval, PDOA will update $y_i(t)$ to be 1 to maximize the dual objective function; otherwise, PDOA will update $y_i(t)$ to $c-c\cdot M$ to ensure that when the threshold is achieved ($M\geq 1$), the sum of all the dual variables in the current ACK interval is exactly c.

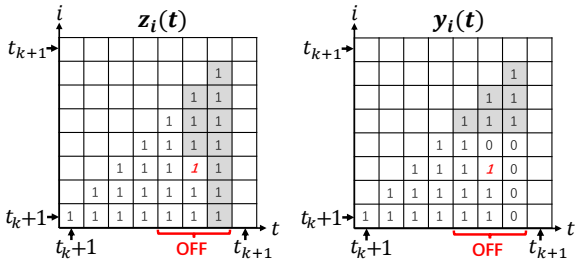
In addition, at the end of slot t (i.e., Lines 19-27), if the channel is ON at slot t, PDOA will skip slot t and go to slot $t+1$. Otherwise, assuming that the most recent ON slot is slot $t^\dagger$ ($t^\dagger\in[L,t)$), then the channels are OFF during $[t^\dagger+1, t]$. To maximize the dual objective function, PDOA updates the dual variables of packet $t^\dagger+1$ to packet t to be 1 since the channels are OFF during $[t^\dagger+1, t]$ and the updating of their dual variables does not violate constraint (7b) (see an illustration in Fig. 1(b)). Note that there may be some OFF channels before slot $t^\dagger$, but PDOA does not update their dual variables to avoid the violation of constraint (7b). For example, assuming that slot t' ($t'<t^\dagger$) is an OFF channel and we let $y_{t'}(t)=1$. Letting $y_{t'}(t)=1$ has no effect on the constraint (7b) at slot t' since we always have $s(t')\sum_{i=1}^{\hat{t}}\sum_{\tau=t'}^{T} y_i(\tau)=0$, but doing this does impact the constraint (7b) at slot $t^\dagger$ (i.e., increasing $s(t^\dagger)\sum_{i=1}^{t^\dagger}\sum_{\tau=t^\dagger}^{T} y_i(\tau)$ by 1 because $y_{t'}(t)$ is a part of $s(t^\dagger)\sum_{i=1}^{t^\dagger}\sum_{\tau=t^\dagger}^{T} y_i(\tau)$ as $t'<t^\dagger$ and $t^\dagger\leq t$), possibly making constraint (7b) at slot $t^\dagger$ violated.

Algorithm 1: Primal-dual-based Online Algorithm (PDOA)

```
   Input : c, s (revealed in an online manner)
   Output: d(t), z_i(t), y_i(t)
   Init.: d(t), z_i(t), y_i(t), L, M ← 0 for all i and t
1  for t = 1 to T do
       /* Iterate all the packets arriving
          since the latest ACK time L.          */
2      for i = L + 1 to t do
3          if M < 1 then      /* Not ready to ACK */
4              z_i(t) ← 1;
5              M ← M + 1/c;
6              y_i(t) ← min{1, c − c · M};
7          end
8          if M ≥ 1 then            /* Ready to ACK */
9              if s(t) = 1 then        /* ON channel */
10                 d(t) ← 1;
11                 M ← 0;
12                 L ← t;
13                 break and go to the next slot (i.e., t + 1);
14             else                    /* OFF channel */
15                 z_i(t) ← 1;
16             end
17         end
18     end
       /* At the end of slot t, update dual
          variable y_i(t) with s(i) = 0 as:     */
19     for i = t decrease to L + 1 do
20         if s(i) = 0 then
21             if y_i(t) = 0 then
22                 y_i(t) ← 1;
23             end
24         else
25             break and go to the next slot;
26         end
27     end
28 end
```

Theorem 1. *PDOA is 3-competitive.*

Proof sketch. We first show that given any channel state $\mathbf{s}$, PDOA produces a feasible solution to primal Problem (6) and dual Problem (7). Then, we show that in any k-th ACK interval, the ratio between the primal objective value and the dual objective value (denoted by $P(k)$ and $D(k)$, respectively) is at most 3, i.e., $P(k)/D(k)\leq 3$. This implies that the ratio between the total primal objective value (denoted by P) and the total dual objective value (denoted by D) is also at most 3,



(a) Primal variables $z_i(t)$ updates. (b) Dual variables $y_i(t)$ updates.

Fig. 1. The updates of primal variables $z_i(t)$ and dual variables $y_i(t)$ in the k-th ACK interval $[t_k+1, t_{k+1}]$, where channels are OFF during $[t_k+5, t_k+7]$. The x-axis represents time and the y-axis represents the packet id. Two ACKs are made at the ON slot t_k and the ON slot t_{k+1}, where the ACK cost $c = 18$. The primal variables $z_i(t)$ and dual variables $y_i(t)$ are updated from slot t_k+1 to slot t_{k+1}; and in slot t, packets are updated from packet t_k+1 to packet t. The red bold italic 1 denotes when the ACK marker equals or is larger than 1. In Fig. 1(a), the grey areas denote the updates due to Line 15; In Fig. 1(b), the grey areas denote the updates due to Lines 19-27.

i.e., $P/D \leq 3$. By the weak duality, PDOA is 3-competitive.

Remark 1. *In [6], the authors also propose a primal-dual-based online algorithm and show that their algorithm achieves a CR of $e/(e-1)$. However, this CR is achieved only in an (unrealistic) asymptotic setting (i.e., when the transmission cost c goes to infinity but the time horizon T is finite). Specifically, in their algorithm, to maximize the dual objective function, at the end of the last slot T, they update certain dual variables $y_i(t)$ that arrived at the OFF slot to be* 1. *In their analysis (the proof of their Theorem* 7*), they show that because of those dual variables updates at the end of slot T, the primal objective value satisfies $P \leq (1+1/((1+1/c)^{\lfloor c \rfloor}-1)) \cdot D + (T(T+1)/2) \cdot (D/c)$. When c goes to infinity, their CR becomes $P/D \leq e/(e-1)$ as $(T(T+1)/2)/c$ goes to* 0 *since T is finite. However, the optimization problem becomes trivial in this setting since an optimal algorithm is simply not to transmit at all given that the transmission cost c can significantly exceed the total staleness cost (at most $(T(T+1)/2)$). Furthermore, when c is finite, their CR is a quadratic function of the time horizon T, which can be very large when T is large. Instead, our analysis holds for any T and c. In our algorithm, rather than updating these dual variables $y_i(t)$ at the end of slot T, we directly update them only in the current ACK interval (i.e., Lines 19-27), ensuring that the dual constraint* (7b) *is satisfied and the dual objective function is as large as possible. This enables us to focus on the analysis of $P(k)/D(k)$ in the current ACK interval and show that $P(k)/D(k) \leq 3$ for any k-th ACK interval, which implies that PDOA is 3-competitive.*

V. Learning-augmented Online Algorithm

Online algorithms are known for their robustness against worst-case scenarios, but they can be overly conservative and may have a poor average performance in typical scenarios. In contrast, ML algorithms leverage historical data to train models that excel in average cases. However, they typically lack worst-case performance guarantees when facing distribution shifts or outliers. To attain the best of both worlds, we design a learning-augmented online algorithm that achieves both consistency and robustness.

A. Machine Learning Predictions

We consider the case where an ML algorithm provides a prediction $\mathcal{P} \triangleq \{p_1, p_2, \ldots, p_n\}$ that represents the times to transmit an ACK for the destination (i.e., the prediction $\mathcal{P}$ makes a total of n ACKs and sends the i-th ACK at slot p_i). The prediction $\mathcal{P}$ is unaware of the channel state pattern $\mathbf{s}$ and can be provided either in full in the beginning (i.e., $t = 0$) or be provided one-by-one in each slot. Furthermore, when the prediction $\mathcal{P}$ decides to send an ACK at an OFF slot, we will simply ignore the decision for this particular slot.

Provided with the prediction $\mathcal{P}$, we specify a trust parameter $\lambda \in (0, 1]$ to reflect our confidence in the prediction: a smaller λ means higher confidence. The learning-augmented online algorithm takes a prediction $\mathcal{P}$, a trust parameter λ, and a channel state pattern $\mathbf{s}$ (revealed in an online manner) as inputs, and outputs a solution with a cost of $C(\mathbf{s}, \mathcal{P}, \lambda)$. A learning-augmented algorithm is said $\beta(\lambda)$-*robust* ($\beta(\lambda) \geq 1$) and $\gamma(\lambda)$-*consistent* ($\gamma(\lambda) \geq 1$) if its cost satisfies

$$C(\mathbf{s}, \mathcal{P}, \lambda) \leq \min\{\beta(\lambda) \cdot OPT(\mathbf{s}), \gamma(\lambda) \cdot C(\mathbf{s}, \mathcal{P})\}, \quad (8)$$

where $OPT(\mathbf{s})$ and $C(\mathbf{s}, \mathcal{P})$ is the cost of the optimal offline algorithm and the cost of purely following the prediction $\mathcal{P}$ under the channel state pattern $\mathbf{s}$, respectively.

We aim to design a learning-augmented online algorithm for primal Problem (6) that exhibits two desired properties (i) *consistency*: when the ML prediction $\mathcal{P}$ is accurate ($C(\mathbf{s}, \mathcal{P}) \approx OPT(\mathbf{s})$) and we trust it, our learning-augmented online algorithm should perform closely to the optimal offline algorithm (i.e., $\gamma(\lambda) \to 1$ as $\lambda \to 0$); and (ii) *robustness*: even if the ML prediction $\mathcal{P}$ is inaccurate, our learning-augmented online algorithm still retains a worst-case guarantee (i.e., $C(\mathbf{s}, \mathcal{P}, \lambda) \leq \beta(\lambda) \cdot OPT(\mathbf{s})$ for any prediction $\mathcal{P}$).

B. Learning-augmented Online Algorithm Description

We present our Learning-augmented Primal-dual-based Online Algorithm (LAPDOA) in Algorithm 2. LAPDOA behaves similarly to PDOA, but the updates of primal variables and dual variables incorporate the ML prediction $\mathcal{P}$.

In LAPDOA, two additional auxiliary variables M' and y' are used to denote the increment of the ACK marker M and the increment of the dual variables $y_i(t)$ in each iteration of update, respectively. Assuming that the current time is t, let $\alpha(t)$ denote the next time when the prediction $\mathcal{P}$ sends an ACK (i.e., $\alpha(t) \triangleq \min\{p_i : p_i \geq t\}$ and $\alpha(t) = \infty$ if $t > p_n$). For the updates of primal and dual variables of an unacked packet i at slot t, based on the relationship between the current time t and $\alpha(i)$ (which is also the time when the prediction $\mathcal{P}$ makes an ACK for packet i because packet i arrives at slot i), we classify them into three types:

- Big updates: those updates make $M' \leftarrow 1/\lambda c$, $y' \leftarrow 1$, and $z_i(t) \leftarrow 1$. The big updates are made when LAPDOA is behind the ACK scheduled by the prediction $\mathcal{P}$ (i.e.,

Algorithm 2: Learning-augmented Primal-dual-based Online Algorithm (LAPDOA)

Input : c, $\mathcal{P}$, λ, $\mathbf{s}$ (revealed in an online manner)
Output: $d(t), z_i(t), y_i(t)$
Init.: $d(t), z_i(t), y_i(t), L, M \leftarrow 0$ for all i and t

for $t = 1$ **to** T **do**
  /* Iterate all the packets arriving since the most recent ACK time L. */
  for $i = L+1$ **to** t **do**
    if $M < 1$ **then** /* Not ready to ACK */
      if $t \geq \alpha(i)$ **then**
        /* Big update: prediction already acked packet i */
        $M' \leftarrow 1/\lambda c$, $y' \leftarrow 1$;
      else
        /* Small update: prediction did not ack packet i yet */
        $M' \leftarrow \lambda/c$, $y' \leftarrow \lambda$;
      end
      $z_i(t) \leftarrow 1$;
      $M \leftarrow M + M'$;
      $y_i(t) \leftarrow y'$;
    end
    if $M \geq 1$ **then** /* Ready to ACK */
      if $s(t) = 1$ **then** /* ON channel */
        $d(t) \leftarrow 1$;
        $M \leftarrow 0$;
        $L \leftarrow t$;
        break and go to the next slot (i.e., $t+1$);
      else /* OFF channel */
        if $z_i(t) \neq 1$ **then**
          /* Zero update */
          $z_i(t) \leftarrow 1$;
        end
      end
    end
  end
  /* At the end of slot t, update dual variable $y_i(t)$ with $s(i) = 0$ as: */
  for $i = t$ ***decrease*** **to** $L+1$ **do**
    if $s(i) = 0$ **then**
      if $y_i(t) = 0$ **then**
        $y_i(t) \leftarrow 1$;
      end
    else
      break and go to the next slot;
    end
  end
end

$t \geq \alpha(i)$), and it tries to catch up the prediction $\mathcal{P}$ by making a big increase in the ACK marker.

- Small updates: those updates make $M' \leftarrow \lambda/c$, $y' \leftarrow \lambda$, and $z_i(t) \leftarrow 1$. The small updates are made when LAPDOA is ahead of the ACK scheduled by the prediction $\mathcal{P}$ (i.e., $t < \alpha(i)$), and LAPDOA tries to slow down its ACK rate by making a small increase in the ACK marker.
- Zero updates: those updates make $M' \leftarrow 0$, $y' \leftarrow 0$, and $z_i(t) \leftarrow 1$. The zero updates are made when LAPDOA is supposed to ACK at some slot t' but finds that slot t' is OFF, and it has to delay its ACK to the next ON slot and pay the holding cost (i.e., $z_i(t) = 1$) along the way.

C. Learning-augmented Online Algorithm Analysis

In this subsection, we focus on the consistency and robustness analysis of LAPDOA with $\lambda \in (0, 1]$. The special cases of LAPDOA with $\lambda = 0$ and $\lambda = 1$ correspond to the cases that LAPDOA follows the prediction $\mathcal{P}$ purely and PDOA, respectively. It is noteworthy that by choosing different values of λ, LAPDOA exhibits a crucial trade-off between consistency and robustness.

Theorem 2. *For any channel state pattern* $\mathbf{s}$*, any prediction* $\mathcal{P}$*, any parameter* $\lambda \in (0, 1]$*, and any ACK cost* c*, LAPDOA outputs an almost feasible solution (within a factor of* $c/(c+1)$*) with a cost of: when* $\lambda \in (0, 1/c]$*,*

$$C(\mathbf{s}, \mathcal{P}, \lambda) \leq \min\{(3/\lambda) \cdot ((c+1)/c) \cdot OPT(\mathbf{s}), (1+\lambda)C_H(\mathbf{s}, \mathcal{P}) + C_A(\mathbf{s}, \mathcal{P})\}, \quad (9)$$

and when $\lambda \in (1/c, 1]$*,*

$$C(\mathbf{s}, \mathcal{P}, \lambda) \leq \min\{(3/\lambda) \cdot ((c+1)/c) \cdot OPT(\mathbf{s}), (\lambda+2)C_H(\mathbf{s}, \mathcal{P}) + (1/\lambda + 2) \cdot \lceil \lambda c \rceil \cdot C_A(\mathbf{s}, \mathcal{P})/c\}, \quad (10)$$

where $C_A(\mathbf{s}, \mathcal{P})$ *and* $C_H(\mathbf{s}, \mathcal{P})$ *denote the total ACK costs and total holding costs of prediction* $\mathcal{P}$ *under* $\mathbf{s}$*, respectively.*

Next, we show that LAPDOA has the robustness guarantee in Lemma 2 and the consistency guarantee in Lemma 3. Combining Lemmas 2 and 3, we can conclude Theorem 2.

Lemma 2. *(Robustness) For any ON/OFF input instance* $\mathbf{s}$*, any prediction* $\mathcal{P}$*, any parameter* $\lambda \in (0, 1]$*, and any ACK cost* c*, LAPDOA outputs a solution which has a cost of*

$$C(\mathbf{s}, \mathcal{P}, \lambda) \leq (3/\lambda) \cdot ((c+1)/c) OPT(\mathbf{s}). \quad (11)$$

Proof sketch. We first show that LAPDOA produces a feasible primal solution and an almost feasible dual solution (with a factor of $c/(c+1)$). Then, we show that in any k-th ACK interval, LAPDOA achieves $P(k)/D(k) \leq 3/\lambda$. This implies that LAPDOA also achieves $P/D \leq 3/\lambda$ on the entire instance. Finally, by scaling down all dual variables $y_i(t)$ generated by LAPDOA by a factor of $c/(c+1)$, we obtain a feasible dual solution with a dual objective value of $(c/(c+1)) \cdot D$. By the weak duality, we have $P/OPT \leq P/((c/(c+1)) \cdot D) = (P/D) \cdot ((c+1)/c) \leq (3/\lambda) \cdot ((c+1)/c)$.

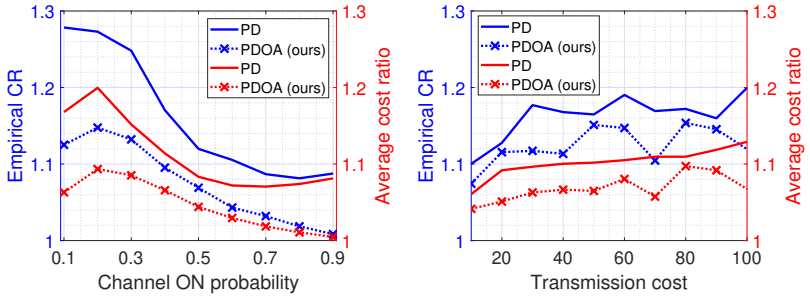
Lemma 3. *(Consistency) For any channel state pattern* $\mathbf{s}$*, any prediction* $\mathcal{P}$*, any parameter* $\lambda \in (0, 1]$*, and any ACK cost* c*, LAPDOA outputs a solution with a cost of: when* $\lambda \in (0, 1/c]$*,*

$$C(\mathbf{s}, \mathcal{P}, \lambda) \leq (1+\lambda)C_H(\mathbf{s}, \mathcal{P}) + C_A(\mathbf{s}, \mathcal{P}), \quad (12)$$

and when $\lambda \in (1/c, 1]$*,*

$$C(\mathbf{s}, \mathcal{P}, \lambda) \leq (\lambda+2)C_H(\mathbf{s}, \mathcal{P}) + (1/\lambda + 2) \cdot \lceil \lambda c \rceil \cdot C_A(\mathbf{s}, \mathcal{P})/c, \quad (13)$$

where $C_A(\mathbf{s}, \mathcal{P})$ *and* $C_H(\mathbf{s}, \mathcal{P})$ *denote the total ACK cost and total holding cost of prediction* $\mathcal{P}$ *under* $\mathbf{s}$*, respectively.*



(a) Bernoulli process (b) Real trace dataset

Fig. 2. Performance comparison of online algorithms under different datasets.

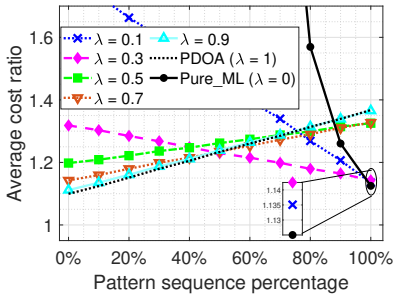


Fig. 3. Performance comparison of learning-augmented algorithms under different trust parameters using synthetic dataset.

Proof sketch. In general, LAPDOA generates three types of updates: big updates, small updates, and zero updates. Our idea is to bound the total cost of each type of update by the cost of the algorithm that purely follows the prediction. In the case of $\lambda \in (0, 1/c]$, we show that the total number of big updates is $C_A(\mathbf{s}, \mathcal{P})/c$, and each big update increases the primal objective value by c, so the total cost of big updates in LAPDOA is $C_A(\mathbf{s}, \mathcal{P})$. In addition, the total number of small updates and zero updates can be shown to be bounded by $C_H(\mathbf{s}, \mathcal{P})$, and each small update or zero update incurs a cost at most $1+\lambda$, thus the total cost of small and zero updates in LAPDOA is at most $(1+\lambda)C_H(\mathbf{s}, \mathcal{P})$. In summary, the total cost of LAPDOA in the case of $\lambda \in (0, 1/c]$ is bounded by $(1+\lambda)C_H(\mathbf{s}, \mathcal{P}) + ((c+1)/c)C_A(\mathbf{s}, \mathcal{P})$. A similar bound can be also obtained for the case of $\lambda \in (1/c, 1]$.

Remark 2. *When we trust the ML prediction (i.e., $\lambda \to 0$) and the ML prediction is accurate at the same time ($C(\mathbf{s}, \mathcal{P}) \approx OPT(\mathbf{s})$), our learning-augmented algorithm also performs nearly to the optimal offline algorithm, achieving consistency.*

Remark 3. *With any $\lambda \in (0, 1]$, the CR of LAPDOA is at most $(3/\lambda) \cdot ((c+1)/c)$, regardless of the prediction quality. This indicates that our learning-augmented algorithm has the worst-case performance guarantees, achieving robustness.*

VI. Numerical Results

In this section, we perform simulations using both synthetic data and real trace data to show that our online algorithm PDOA outperforms the State-of-the-Art online algorithm and that our learning-augmented online algorithm LAPDOA achieves consistency and robustness.

A. Online Algorithm

In Fig. 2, we compare PDOA with the State-of-the-Art online algorithm proposed in [6] (which is referred to as "PD" in Fig. 2). Two datasets are considered: (i) The synthetic dataset in Fig. 2(a). We adopt the same settings as in [6], where the channel state is a Bernoulli process with varying channel ON probability and the transmission cost $c = 15$; (ii) The real trace dataset [24] in Fig. 2(b). This dataset contains the channel measurement (i.e., reference signal received quality (RSRQ)) for commercial mmWave 5G services in downtown Minneapolis. Researchers conducted walking tests in a 1300m loop area using an Android smartphone with a 5G monitoring tool to collect RSRQ data. We set a threshold (-13dB) to determine if the channel is ON or OFF based on whether RSRQ surpasses it or not. Here we vary the transmission cost from 10 to 100. In both datasets, the performance metrics are the empirical CR and the average cost ratio (i.e., the worst cost ratio and the average cost ratio under the online algorithm and the optimal offline algorithm over multiple simulation runs).

Fig. 2 illustrates that our online algorithm PDOA consistently outperforms the State-of-the-Art online algorithm PD in both datasets, i.e., PDOA achieves a lower empirical CR and a lower average cost ratio compared to PD. In addition, the empirical CR of PDOA outperforms the theoretical analysis (with a CR of 3), validating our theoretical results.

B. Learning-augmented Online Algorithm

In this subsection, we study the performance of LAPDOA under different prediction qualities using synthetic datasets. We first explain how to generate ML predictions based on the training dataset. Then, we shift the distribution of the testing dataset to deviate from the training dataset and showcase the performance of LAPDOA on the testing datasets.

In Fig. 2(a), PDOA demonstrates strong performance under the Bernoulli process. However, a specific training dataset reveals its suboptimal performance. In this training dataset, the transmission cost $c = 16$, and the channel state sequence is constituted by an independently repeating pattern [$X\times$OFF, $Y\times$ON], where $X \sim B(13, 0.9)$ and $Y \sim B(6, 0.9)$ ($B(n, p)$ represents the binomial distribution with parameters n and p). Under this pattern, in most cases, PDOA only makes one transmission at the first ON slot of these Y ON slots (i.e., after a long consecutive X OFF slots, the ACK marker M will be larger than 1, and PDOA will transmit at the first ON slot. However, after this transmission, during the short remaining $(Y-1)$ ON slots, the ACK marker M is unable to be increased to 1). This results in a high AoI increase for these next X OFF slots. For the optimal offline algorithm, to have a lower AoI increase during OFF slots, it transmits at both the first and the last ON slot among those Y ON slots. To generate a sequence of channel states of the required length, we repeat the pattern enough times independently and concatenate them together.

Recall that LAPDOA incorporates an ML prediction $\mathcal{P}$ that provides the transmission decision at each slot. To generate such an ML prediction $\mathcal{P}$, we train an *Long Short-term Memory (LSTM)* network, which has three LSTM layers (each layer has 20 hidden states) followed by one fully connected layer. The input of our LSTM network is the current channel state, and the output is the transmission probability at that slot. For training, we manually create 300 sequences, each with a length of 100 slots consisting of repeating patterns introduced earlier (we call these constructed sequences "pattern sequences"). Optimal offline transmission decisions for the training datasets are obtained through dynamic programming. We use the mean squared error between the LSTM network output and the optimal offline algorithm output as the loss function and employ the Adam optimizer to train the weights. In the end, to convert the output of our LSTM network (i.e., transmission probability) to the real transmission decisions, a threshold (e.g., 0.5) is set, and transmission occurs when the output of the LSTM network exceeds the threshold.

In Fig. 3, we illustrate LAPDOA's performance under varying prediction qualities, influenced by a distribution shift between the training and testing datasets. The training dataset only contains the sequences that are fully composed of the pattern (i.e., the percentage of the pattern sequences is 100%). However, in the testing dataset, we reduce the percentage of the pattern sequence by replacing some pattern sequences with a Bernoulli process sequence of a length of 100 with an ON probability of 0.32 (close to the pattern ON probability). While the training dataset and the testing dataset share the same channel ON probability, they exhibit variations in distribution. The magnitude of this shift amplifies as the percentage of the pattern sequence decreases. As we can observe in Fig. 3, when the distribution shift is small (100% or 90% pattern sequence percentage), our trained ML algorithm ("Pure_ML" in the figure) outperforms PDOA. Learning-augmented algorithms trusting the prediction ($\lambda \in \{0.1, 0.3\}$) closely match the ML algorithm's performance. Conversely, with a substantial distribution shift (0 or 10% pattern sequence percentage), the ML algorithm performs poorly while PDOA performs well. In this case, learning-augmented algorithms not trusting the prediction ($\lambda \in \{0.7, 0.9\}$) closely resemble PDOA. Furthermore, with different values of λ, LAPDOA provides different tradeoff curves for consistency and robustness.

VII. Conclusion

In this paper, we studied the minimization of data freshness and transmission costs under a time-varying wireless channel. After reformulating our original problem to a TCP ACK problem, we developed a 3-competitive primal-dual-based online algorithm. Realizing the pros and cons of online algorithms and ML algorithms, we designed a learning-augmented online algorithm that takes advantage of both approaches and achieves consistency and robustness. Finally, simulation results validate the superiority of our online algorithm and highlight the consistency and robustness achieved by our learning-augmented algorithm. For future work, one interesting direction would be to consider how to adaptively select the trust parameter λ to achieve the best performance.

References

[1] S. Li, L. D. Xu, and S. Zhao, "The internet of things: a survey," *Information systems frontiers*, vol. 17, pp. 243–259, 2015.

[2] F. Wu, C. Rüdiger, and M. R. Yuce, "Real-time performance of a self-powered environmental iot sensor network system," *Sensors*, vol. 17, no. 2, p. 282, 2017.

[3] X. Cao, J. Wang, Y. Cheng, and J. Jin, "Optimal sleep scheduling for energy-efficient aoi optimization in industrial internet of things," *IEEE Internet of Things Journal*, vol. 10, no. 11, pp. 9662–9674, 2023.

[4] B. Yu, Y. Cai, X. Diao, and K. Cheng, "Adaptive packet length adjustment for minimizing age of information over fading channels," *IEEE Transactions on Wireless Communications*, pp. 1–1, 2023.

[5] S. Kaul, R. Yates, and M. Gruteser, "Real-time status: How often should one update?" in *2012 IEEE INFOCOM*, 2012, pp. 2731–2735.

[6] Y.-H. Tseng and Y.-P. Hsu, "Online energy-efficient scheduling for timely information downloads in mobile networks," in *2019 ISIT*, 2019, pp. 1022–1026.

[7] A. R. Karlin, C. Kenyon, and D. Randall, "Dynamic tcp acknowledgement and other stories about e/(e-1)," in *Proceedings of the thirty-third annual ACM symposium on Theory of computing*, 2001, pp. 502–509.

[8] Z. Liu, K. Zhang, B. Li, Y. Sun, Y. T. Hou, and B. Ji, "Learning-augmented online minimization of age of information and transmission costs," *arXiv preprint arXiv:2403.02573*, 2024.

[9] Y. Sun, I. Kadota, R. Talak, and E. Modiano, *Age of information: A new metric for information freshness*. Springer Nature, 2022.

[10] R. D. Yates, Y. Sun, D. R. Brown, S. K. Kaul, E. Modiano, and S. Ulukus, "Age of information: An introduction and survey," *IEEE JSAC*, vol. 39, no. 5, pp. 1183–1210, 2021.

[11] E. Fountoulakis, N. Pappas, M. Codreanu, and A. Ephremides, "Optimal sampling cost in wireless networks with age of information constraints," in *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2020, pp. 918–923.

[12] Z. Liu, B. Li, Z. Zheng, Y. T. Hou, and B. Ji, "Towards optimal tradeoff between data freshness and update cost in information-update systems," *IEEE Internet of Things Journal*, pp. 1–1, 2023.

[13] K. Saurav and R. Vaze, "Minimizing the sum of age of information and transmission cost under stochastic arrival model," in *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications*, 2021, pp. 1–10.

[14] A. M. Bedewy, Y. Sun, S. Kompella, and N. B. Shroff, "Optimal sampling and scheduling for timely status updates in multi-source networks," *IEEE TIT*, vol. 67, no. 6, pp. 4019–4034, 2021.

[15] A. Sinha and R. Bhattacharjee, "Optimizing age-of-information in adversarial and stochastic environments," *IEEE Transactions on Information Theory*, vol. 68, no. 10, pp. 6860–6880, 2022.

[16] S. Banerjee and S. Ulukus, "Age of information in the presence of an adversary," in *IEEE INFOCOM 2022 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2022, pp. 1–8.

[17] S. Li, C. Li, Y. Huang, B. A. Jalaian, Y. T. Hou, and W. Lou, "Enhancing resilience in mobile edge computing under processing uncertainty," *IEEE JSAC*, vol. 41, no. 3, pp. 659–674, 2023.

[18] T. Lykouris and S. Vassilvitskii, "Competitive caching with machine learned advice," *J. ACM*, vol. 68, no. 4, jul 2021.

[19] M. Purohit, Z. Svitkina, and R. Kumar, "Improving online algorithms via ml predictions," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[20] E. Bamas, A. Maggiori, and O. Svensson, "The primal-dual method for learning augmented algorithms," *Advances in Neural Information Processing Systems*, vol. 33, pp. 20 083–20 094, 2020.

[21] D. Rutten, N. Christianson, D. Mukherjee, and A. Wierman, "Smoothed online optimization with unreliable predictions," *Proc. ACM Meas. Anal. Comput. Syst.*, vol. 7, no. 1, mar 2023.

[22] N. Buchbinder, J. S. Naor *et al.*, "The design of competitive online algorithms via a primal–dual approach," *Foundations and Trends® in Theoretical Computer Science*, vol. 3, no. 2–3, pp. 93–263, 2009.

[23] S. P. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[24] A. Narayanan, E. Ramadan, R. Mehta, X. Hu, Q. Liu, R. A. Fezeu, U. K. Dayalan, S. Verma, P. Ji, T. Li *et al.*, "Lumos5g: Mapping and predicting commercial mmwave 5g throughput," in *Proceedings of the ACM Internet Measurement Conference*, 2020, pp. 176–193.